

# Perancangan dan Implementasi Sistem Pengaturan Optimal $LQR$ untuk Menjaga Kestabilan Hover pada *Quadcopter*

Kardono, Rusdhianto Effendi AK, dan Ali Fatoni

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

E-mail: [fatoni@ee.its.ac.id](mailto:fatoni@ee.its.ac.id)

**Abstrak**—*Quadcopter* adalah pesawat terbang yang memiliki potensi untuk lepas landas, *hover*, terbang *manuver*, dan mendarat bahkan di daerah kecil. Seiring dengan perkembangan teknologi modern, saat ini *quadcopter* banyak digunakan untuk pengawasan area, pengambilan foto/video, pelaksanaan misi yang beresiko tinggi dan lain-lain. Kestabilan *hover* pada *quadcopter* sangatlah penting dan harus dimiliki *quadcopter* agar pemanfaatannya dapat maksimal. Kontrol *hover* merupakan prioritas utama dalam setiap upaya pengendalian *quadcopter* baik pada pengendalian fase *take-off*, *landing*, dan *trajectory*, hal ini dikarenakan kesalahan yang kecil saja yang terjadi pada sudut dan atau ketinggian *quadcopter* dapat menyebabkan *quadcopter* bergerak baik terhadap sumbu  $x$ ,  $y$ , maupun  $z$ .

Pada Tugas Akhir ini, didapatkan nilai parameter kontrol  $LQR$  dari hasil *tuning* diperoleh parameter  $R=1$  dan  $Q=Q_4$  yang pada simulasi dapat terbang *hover* pada ketinggian 2 m, dan dapat mengatasi gangguan dengan *rise time* selama 0,1332 detik. Respon hasil implementasi pada *quadcopter* tidak sebaik dengan hasil simulasi, terbang *hover* dengan *set point* ketinggian 100 cm masih berisikasi antara 50cm sampai 200cm, dan respon kestabilan sudut lebih lambat yaitu 0,23detik.

**Kata Kunci**—*Quadcopter*, *Hover*,  $LQR$ , Kontrol Optimal.

## I. PENDAHULUAN

Penelitian yang saat ini berkembang adalah mengenai *auto pilot* atau bahkan sampai pada pesawat tanpa awak. Hal tersebut terlihat dari banyak ditemuinya penelitian berkenaan dengan pengendalian gerakan-gerakan *quadcopter*. Penelitian pesawat tanpa awak banyak dikemukakan untuk berbagai tujuan, di antaranya adalah untuk uji coba pesawat pengambilan foto atau video, pelaksanaan misi yang beresiko tinggi dan lain-lain.

Sebuah *quadcopter* adalah kendaraan yang memiliki potensi untuk lepas landas, *hover*, terbang *manuver*, dan mendarat bahkan di daerah kecil dan memiliki mekanisme kontrol sederhana. Namun, *quadcopter* adalah sistem kompleks yang tidak stabil dan dapat menjadi sulit untuk terbang tanpa *embedded* sistem kontrol.

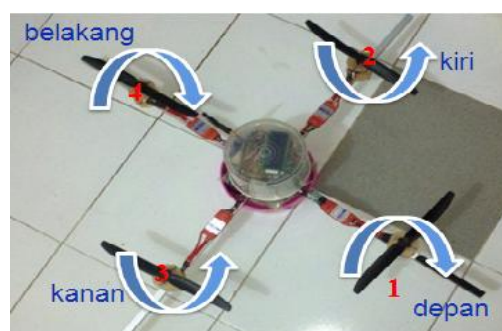
Dalam pemanfaatan *quadcopter* untuk berbagai tujuan, kestabilan *hover* pada *quadcopter* sangatlah penting dan harus dimiliki *quadcopter* agar pemanfaatannya dapat maksimal.

Penelitian pengendalian *quadcopter* agar dapat bergerak *hover* dengan stabil dengan menggunakan metode kontrol optimal *Linier Quadratic Regulator (LQR)*, masih merupakan topik yang sangat menarik untuk diteliti.

## II. TEORI PENUNJANG

### A. Pergerakan *Quadcopter*

*Quadcopter* adalah pengembangan dari *helicopter* yang hanya memiliki sebuah *rotor*, teknologi yang terdapat pada *quadcopter* ini menggunakan sinkronisasi antara keempat *rotor* yang dikonfigurasi dalam bentuk *frame plus (+)* dimana *rotor* depan dan belakang berputar searah jarum jam, sedangkan *rotor* sebelah kanan dan kiri bergerak berlawanan arah jarum jam. *Quadcopter* dapat melakukan *take off* dan *landing* secara vertikal.

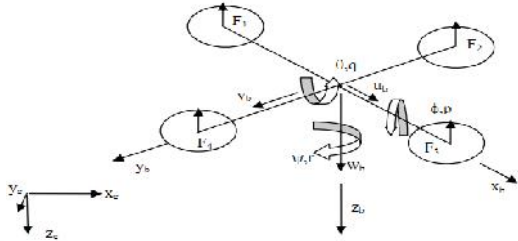


Gambar 1. Bentuk *Quadcopter*

Dua pasang baling-baling (1,4) dan (2,3) berputar dengan arah yang berlawanan. Dengan memvariasikan kecepatan *rotor*, dapat mengubah gaya angkat dan menciptakan gerak. Dengan demikian, meningkatkan atau menurunkan kecepatan keempat baling-baling bersama-sama menghasilkan gerakan vertikal. Mengubah kecepatan baling-baling 2 (kiri) dan 3 (kanan) akan menghasilkan gerakan *rotasi roll* ditambah dengan gerakan *translasi* terhadap sumbu  $y$  sedangkan mengubah kecepatan baling-baling 1 (depan) dan 4 (belakang) akan menghasilkan gerakan *rotasi pitch* ditambah dengan gerakan *translasi* terhadap sumbu  $x$ .

### B. Model Quadcopter [1][2]

Quadcopter memiliki 6 *degree of freedom (DOF)*. Untuk mendeskripsikan gerakan dari 6 *DOF rigid-body* digunakan dua buah frame referensi yaitu *earth inertial reference (E-frame)* dan *body fixed reference (B-frame)*



Gambar 2. Ilustrasi B-frame Terhadap E-frame

Persamaan kinematika dari 6-DOF rigid-body

$$\dot{\xi} = J_{\Theta} v \quad (1)$$

$$v = [V^B \omega^B]^T = [u \ v \ w \ p \ q \ r]^T \quad (2)$$

$$J_{\Theta} = \begin{bmatrix} R_{\Theta} & 0_{3 \times 3} \\ 0_{3 \times 3} & T_{\Theta} \end{bmatrix} \quad (3)$$

Matrik rotasi ( $R_{\Theta}$ ) dan matrik translasi ( $T_{\Theta}$ ) ditunjukkan pada persamaan

$$R = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - c\phi\psi & c\psi s\theta c\phi + s\psi s\phi \\ s\phi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\theta c\phi \end{bmatrix} \quad (4)$$

$$T = \begin{bmatrix} 1 & \tan\theta \sin\phi & \tan\theta \cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sec\theta \sin\phi & \sec\theta \cos\phi \end{bmatrix} \quad (5)$$

Dinamika *rigid body* 6 DOF memperhitungkan massa body  $m$  [kg] dan matriks inersi  $I$  [N m s<sup>2</sup>]. Dinamika digambarkan oleh Persamaan 6

$$\begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} \begin{bmatrix} \dot{V}^B \\ \dot{\omega}^B \end{bmatrix} + \begin{bmatrix} \omega^B \times (mV^B) \\ \omega^B \times (I\omega^B) \end{bmatrix} = \begin{bmatrix} F^B \\ \tau^B \end{bmatrix} \quad (6)$$

Dimana notasi  $I_{3 \times 3}$  berarti 3 kali 3 matriks identitas.  $\dot{V}^B$  [ms<sup>-2</sup>] adalah vector linier percepatan quadcopter mengacu B-frame sementara  $\dot{\omega}^B$  [rad s<sup>-2</sup>] adalah vector percepatan sudut quadcopter mengacu B-frame. Selain itu,  $F^B$  [N] adalah vector gaya quadcopter mengacu B-frame dan  $\tau^B$  [N m] adalah vector torsi quadcopter mengacu B-frame.

Gaya yang terjadi pada sistem

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \frac{dv}{dt_e} = m \left( \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \right) \quad (7)$$

Sehingga percepatan linear dari quadcopter dapat dihitung

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} - \begin{bmatrix} qw - rv \\ ru - vw \\ pv - qu \end{bmatrix} \quad (8)$$

Moment yang terjadi

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (9)$$

sehingga

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = I^{-1} \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} - I^{-1} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (10)$$

Inertia matrik adalah

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (11)$$

dimana

$$I_{xx} = \frac{m\rho^2}{4} + \frac{mh^2}{6} + 2ml^2 + \frac{MR^2}{4} + \frac{MH^2}{12} \quad (12)$$

$$I_{yy} = \frac{m\rho^2}{4} + \frac{mh^2}{6} + 2ml^2 + \frac{MR^2}{4} + \frac{MH^2}{12} \quad (13)$$

$$I_{zz} = \frac{MR^2}{2} + 4ml^2 \quad (14)$$

Dengan memasukkan inertia pada Persamaan 10

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{I_{yy}-I_{zz}}{I_{xx}} qr \\ \frac{I_{xx}-I_{zz}}{I_{yy}} pr \\ \frac{I_{xx}-I_{yy}}{I_{zz}} pq \end{bmatrix} + \begin{bmatrix} \frac{1}{I_{xx}} \tau\phi \\ \frac{1}{I_{yy}} \tau\theta \\ \frac{1}{I_{zz}} \tau\psi \end{bmatrix} \quad (15)$$

Gaya yang terjadi pada tiap motor di quadcopter

$$U_B(\cdot) = E^{-2} = \begin{bmatrix} 0 \\ 0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tau\phi \\ \tau\theta \\ \tau\psi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b(\Omega_f^2 + \Omega_r^2 + \Omega_b^2 + \Omega_l^2) \\ bl(-\Omega_f^2 + \Omega_l^2) \\ bl(\Omega_f^2 - \Omega_b^2) \\ d(-\Omega_f^2 + \Omega_r^2 - \Omega_b^2 + \Omega_l^2) \end{bmatrix} \quad (16)$$

Selain gaya yang dihasilkan motor, gravitasi juga mempengaruhi gaya yang terjadi pada quadcopter

$$F_g = R \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} = \begin{bmatrix} mg \sin\theta \\ -mg \sin\phi \cos\theta \\ -mg \cos\phi \cos\theta \end{bmatrix} \quad (17)$$

Efek *gyroscopic* yang dihasilkan oleh rotasi propeler, karena dua dari mereka yang berputar searah jarum jam dan dua lainnya berlawanan, ada ketidakseimbangan secara keseluruhan ketika jumlah aljabar dari kecepatan rotor tidak sama dengan nol. Selain itu jika kecepatan sudut *roll* atau *pitch* juga berbeda dari nol, quadcopter mengalami torsi *gyroscopic* menurut Persamaan

$$O_b(v)\Omega = \left[ -\sum_{k=1}^4 Jtp \left( w^b \times \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} (-1)^k \Omega_k \right) \right] = \left[ Jtp \begin{bmatrix} 0_{3 \times 1} \\ -q \\ p \\ 0 \end{bmatrix} \zeta \right]$$

$$= Jtp \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (18)$$

$O_b(v)\Omega$  adalah propeler *gyroscopic* matriks dan JTP [N m s<sup>2</sup>] atau Jr [N m s<sup>2</sup>] Adalah momen inersia rotasi total sekitar sumbu propeler dihitung dengan Persamaan

$$JTP = \frac{M}{2} r^2 \quad (19)$$

Inertia rotasi total sekitar sumbu propeller terjadi disekitar motor, sehingga M adalah masa motor, sedangkan r adalah jari-jari motor.

Persamaan (20) mendefinisikan keseluruhan kecepatan propeler  $\Omega$  [rad s<sup>-1</sup>] yang digunakan dalam Persamaan 18

$$\Omega = -\Omega_f + \Omega_r - \Omega_b + \Omega_l \quad (20)$$

$$M_H \dot{\xi} + C_H(\xi)\dot{\xi} = G_H + O_H(\xi)\Omega + E_H(\xi)\Omega^2 \quad (21)$$

Dimana  $\xi$  adalah vector akselerasi quadcopter yang terjadi pada H-frame. Sedangkan  $M_H$  adalah matrik inersia system pada H-frame yang sesuai dengan B-frame

$$M_H = M_B = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix} \quad (22)$$

Matrik Coriolis-sentripetalnya adalah

$$C_H = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & -S(I\omega^B) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{zz}r & -I_{yy}q \\ 0 & 0 & 0 & -I_{zz}r & 0 & I_{xx}p \\ 0 & 0 & 0 & I_{yy}q & -I_{xx}p & 0 \end{bmatrix} \quad (23)$$

Matriks gerakan pada  $H$ -frame  $E_H(\cdot)$

$$E_H(\xi) = \begin{bmatrix} R_\theta & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} E_B \Omega^2 = \begin{bmatrix} (s_\psi s_\phi + c_\psi s_\theta c_\phi) U_1 \\ (-c_\psi s_\phi + s_\psi s_\theta c_\phi) U_1 \\ (s_\theta c_\phi) U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (24)$$

Dengan membalik Persamaan 21 diperoleh

$$\dot{\xi} = M_H^{-1}(-C_H(\xi)\xi + G_H + O_H(\xi)\Omega + E_H(\xi)\Omega^2) \quad (25)$$

Sehingga Persamaan model dinamika *quadcopter* menjadi

$$\ddot{x} = (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi) \frac{F}{m} \quad (26)$$

$$\ddot{y} = (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi) \frac{F}{m} \quad (27)$$

$$\ddot{z} = -g + (\cos\phi \cos\theta) \frac{F}{m} \quad (28)$$

$$\dot{p} = \frac{I_{yy}-I_{zz}}{I_{xx}} q r + \frac{J_r}{I_{xx}} q \Omega + \frac{U_2}{I_{xx}} \quad (29)$$

$$\dot{q} = \frac{I_{zz}-I_{xx}}{I_{yy}} p r + \frac{J_r}{I_{yy}} p \Omega + \frac{U_3}{I_{yy}} \quad (30)$$

$$\dot{r} = \frac{I_{xx}-I_{yy}}{I_{zz}} p q + \frac{U_4}{I_{zz}} \quad (31)$$

### C. Persamaan State Model

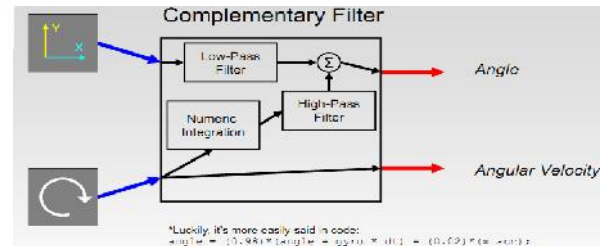
Persamaan state model diperoleh dengan melinierisasi persamaan model *quadcopter* pada Persamaan 26-31 dengan melinierisasi pada *operating point*-nya, dimana *operating point quadcopter* saat terbang hover untuk  $\theta = 0$ ,  $\phi = 0$ ,  $\psi = 0$  sedangkan untuk  $\frac{U_1}{m} = 11.7804$  sehingga diperoleh Persamaan state sistem adalah

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{u_1}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \Omega \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (32)$$

### D. Complementary filter

*Accelerometer* dapat memberikan pengukuran sudut kemiringan yang akurat ketika sistem sedang diam (statis). Bila sistem sedang bergerak, *accelerometer* tidak bisa mengikuti

pergerakan yang cepat dikarenakan responnya yang lambat dan memiliki noise yang besar. Sedangkan pada *gyroscope* membaca kecepatan sudut. Setelah melakukan komputasi integral data dari waktu ke waktu, maka sudut kemiringan dapat dihitung. Tetapi sudut ini akan menjadi tidak akurat dalam jangka panjang karena efek bias yang dihasilkan oleh *gyroscope* dan noise, walaupun noise tersebut sangat kecil. Sehingga dibutuhkan filter yang dapat menggabungkan kedua sensor ini dengan tujuan dapat diperoleh pembacaan kemiringan/sudut yang cepat dan akurat.



Gambar 3. Blok Diagram Komplementari Filter

Beruntungnya, blok diagram tersebut menjadi mudah saat diaplikasikan dalam *code program (discrete)*

$$\text{Angle} = a * (\text{Angle} + \text{gyro} * dt) + (1-a) * (x\_acc); \quad (33)$$

Dimana  $a$  diperoleh dari

$$a = \frac{\tau}{\tau + dt} \quad (34)$$

### E. Kontroler LQR[3]

*Linear Quadratic Regulation*, disebut Linier karena model dan bentuk kontrolernya berupa linier. Sedangkan disebut kuadratik karena cost functionnya adalah kuadratik dan karena referensinya bukanlah berupa fungsi waktu maka disebut regulator.

Suatu persamaan sistem linear :

$$\dot{x} = Ax + Bu \quad (35)$$

$$y = Cx \quad (36)$$

di mana

$x_{n \times 1}$  : State Sistem

$u_{m \times n}$  : Stateinput

$y$  : Stateoutput

$A$  : Matriks Sistem  $A_{n \times n}$

$B$  : Matriks Input  $B_{n \times m}$

$C$  : Matriks Output  $C_{1 \times n}$

Dengan meminimisasi energi (*cost function/quadratic function*) melalui indeks performansi dalam interval  $[t_0, \infty]$  adalah :

$$J = \frac{1}{2} \int_{t_0}^{\infty} (x^T Q x + u^T R u) dt \quad (37)$$

di mana

$t_0$  = waktu awal

$\infty$  = waktu akhir

$Q$  = matriks *semidefinit* positif

$R$  = matriks *definit* positif

Persoalan regulator dapat diselesaikan dengan menyelesaikan Persamaan *Riccati* sebagai berikut :

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (38)$$

Di mana pemilihan pemberat  $Q$  dan  $R$  berpedoman pada :



1. Semakin besar harga Q, semakin memperbesar harga elemen penguatan K sehingga mempercepat sistem untuk mencapai keadaan tunak (*intermediate state cost function*).
2. Semakin besar harga R, maka akan memperkecil harga penguatan K dan memperlambat keadaan tunak (*energy drive*).

di mana :

$$K = R^{-1}B^TP \quad (39)$$

$$u = -Kx \quad (40)$$

### III. PERANCANGAN SISTEM

#### A. Spesifikasi Sistem

*Quadcopter* bisa dikatakan robot terbang yang memiliki 6 derajat kebebasan (*DOF*) yaitu 3 *DOF* pada rotasi dan 3 *DOF* pada translasi. Spesifikasi sistem yang diharapkan tercapai pada perancangan ini adalah sebagai berikut :

1. *Quadcopter* dapat diatur gerakan terbangnya melalui *remote control* yang dikendalikan oleh pilot
2. *Quadcopter* dilengkapi dengan beberapa sensor penunjang gerakan *hover* secara *autonomous*
3. *Quadcopter* dapat menjaga kondisi *hover* ketika mode *autonomous* diaktifkan
4. Data-data sensor dapat diterima *ground station*

#### B. Perancangan dan Implementasi Perangkat Keras

Perancangan perangkat keras pada tugas akhir ini terdiri dari dua bagian yaitu rancang bangun mekanik *quadcopter* dan desain sistem elektronik.

##### 1. Desain Mekanik *Quadcopter*

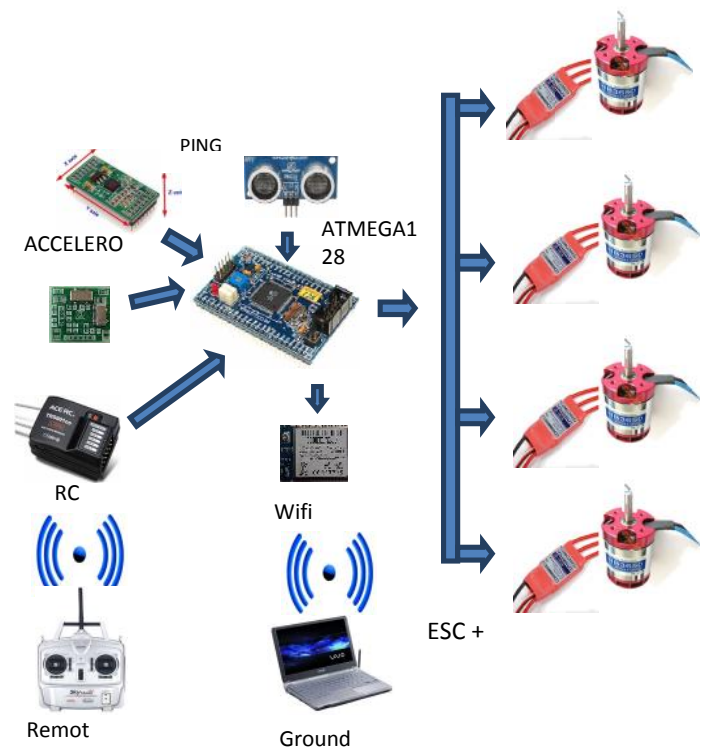
Sistem mekanik yang baik akan mendukung pergerakan *quadcopter* menjadi lebih baik, oleh karena itu perancangan mekanik dalam hal ini *frame* dan kedudukan motor haruslah proporsional dengan titik beban *quadcopter*.



Gambar 4. Hasil Desain Mekanik *Quadcopter*

##### 2. Desain Sistem Elektronik

Sistem elektronika yang ada pada *quadcopter* terdiri atas sistem kontroler yang berupa mikrokontroler dan beberapa sensor yang digunakan sebagai acuan dalam menentukan gerak terbang. Sensor yang digunakan adalah sensor ketinggian menggunakan sensor ultrasonik ping), sensor sudut menggunakan sensor *accelerometer*, dan sensor kecepatan sudut menggunakan sensor *gyroscope*. Selain itu sistem elektronika juga dirancang mampu mengirimkan data ke *ground station*. Perancangan keseluruhan dari sistem elektronika dari *quadcopter* ditunjukkan pada Gambar 5.



Gambar 5 Rancangan Sistem Elektronik *Quadcopter*

#### C. Perancangan Kontroler LQR pada *Quadcopter*

Sinyal kontrol  $u$  yang diberikan ke motor sebagaimana persamaan 40 dengan state yang dipakai adalah  $[z \ V_z \ \phi \ p \ q]^T$ , dimana  $z$  adalah ketinggian yang diperoleh dari pembacaan sensor ping, dan  $V_z$  merupakan kecepatan terhadap sumbu  $z$  yang dapat diperoleh dengan menurunkan  $z$  terhadap waktu yang pada *discrete time* adalah

$$V_z = \frac{z(k) - z(k-1)}{T_s} \quad (41)$$

Sedangkan untuk  $\phi$  dan  $\theta$  diambil dari pembacaan sudut dari sensor *accelerometer*, dan untuk  $p$  dan  $q$  diambil dari pembacaan kecepatan sudut dari sensor *gyroscope*. Sehingga program LQR yang ditanamkan pada *quadcopter* adalah

$$u1 = k_{11} * z + k_{12} \frac{z - z_{before}}{T_s} \quad (42)$$

$$u2 = k_{23} * acc\_roll + k_{24} * gyro\_roll \quad (43)$$

$$u3 = k_{34} * acc\_pitch + k_{35} * gyro\_pitch \quad (44)$$

Tuning matrik pembobot Q dan R dilakukan dengan memberikan pembobot  $R=1$  dan matrik  $Q = C^*C$ , untuk mengatur bobot matrik Q sesuai dengan *state* mana yang lebih diutamakan, semakin besar bobot matrik Q akan mempercepat respon. Gain matrik K dihitung dengan menggunakan library ARE (*Aljabar Riccati Equation*) pada matlab, menghitung gain kontrol K dilakukan dengan mengetik perintah berikut.

$$>>S=ARE(A,B*inv(R)*B',Q); \quad (43)$$

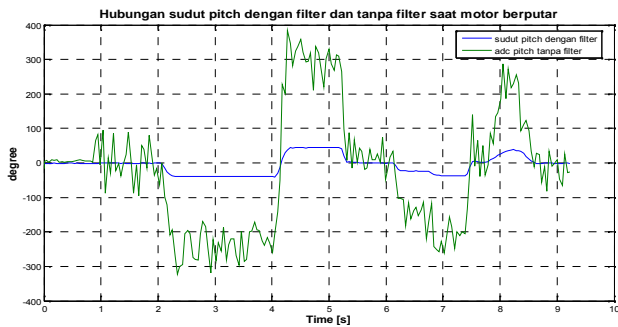
$$>>K=inv(R)*B'*S \quad (44)$$

### IV. PENGUJIAN DAN ANALISA

#### A. Pengujian Complementary Filter

Pengujian dilakukan dengan membandingkan pembacaan *accelerometer* dan hasil *complementary filter*, saat motor

berputar, sehingga ada noise getaran pada *quadcopter* sebagaimana ditunjukkan pada Gambar 6

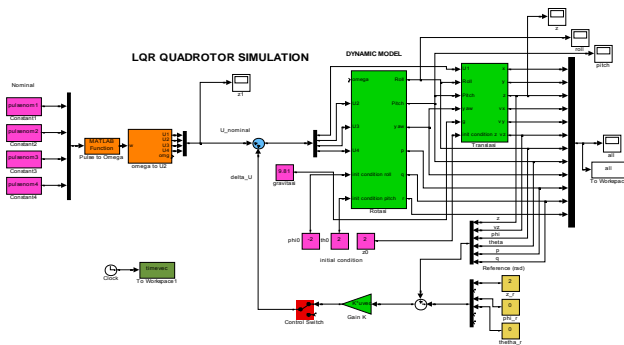


Gambar 6. Pengujian Filter Sudut Saat Motor Berputar

Dari Gambar 6 terlihat bahwa saat motor berputar/dengan getaran, pembacaan *accelerometer* ber-noise, noise pembacaannya cukup besar, sedangkan pembacaan dengan filter terlihat sangat baik, noise dari getaran motor sudah dihilangkan, hal ini menunjukkan penggunaan *filter* telah tepat dan baik.

### B. Simulasi Kontrol LQR– Variasi Bobot Q

Simulasi kontrol LQR dilakukan dengan menggunakan matlab simulink sebagaimana Gambar 7



Gambar 7. Simulasi Kontrol LQR

Dengan mengubah-ubah matrik pembobot Q dan R akan mendapatkan variasi gain K, sebagaimana contoh berikut

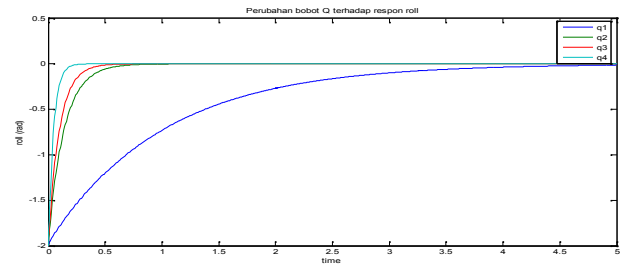
$$Q_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (45)$$

$$Q_2 = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 50 & 0 & 0 & 0 \\ 0 & 0 & 0 & 50 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (46)$$

$$Q_3 = \begin{bmatrix} 50 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (47)$$

$$Q_4 = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 500 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (48)$$

Masing-masing gain matrik K disimulasikan pada matlab simulink, Gambar 8 menunjukka efek dari perubahan pembobot matrik Q



Gambar 8. Perubahan Bobot Q

Berdasarkan respon yang telah ditunjukkan pada gambar 10, *rise time* ( $t_r$ ) dari perubahan bobot Q dengan kriteria 5% - 95% untuk *roll* dan *pitch* besarnya bervariasi

$$\tau = 0.632 * K \quad (49)$$

$$t_r = \tau * \ln 19 \quad (50)$$

Dengan menggunakan Persamaan 49-50, dapat dihitung *rise time* dari masing-masing pembobot Q pada simulasi *roll* dan *pitch* adalah

$$Q_1, \tau = 0.9916 \quad \text{makatr} = 2.919 \text{ s}$$

$$Q_2, \tau = 0.1376 \quad \text{makatr} = 0.405 \text{ s}$$

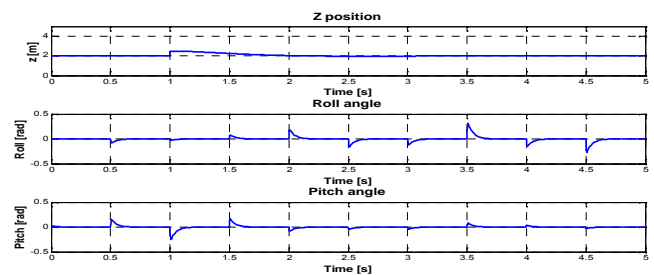
$$Q_3, \tau = 0.09723 \quad \text{makatr} = 0.2862 \text{ s}$$

$$Q_4, \tau = 0.04524 \quad \text{makatr} = 0.1332 \text{ s}$$

Dengan melihat perubahan *rise time* dari masing-masing perubahan pembobot Q terlihat bahwa, semakin besar pembobot Q yang digunakan, maka akan mempercepat respon sistem mencapai keadaan tunaknya (*steady state*).

### C. Simulasi Kontrol LQR – Respon Terhadap Gangguan

Pada simulasi ini, kontroler LQR dengan pembobot Q yang dipilih ( $Q_4$ ) disimulasikan sebagaimana gambar 7 dengan mengaktifkan *switchnoise* untuk *roll*, *pitch* dan *z*, noise *roll* dan *pitch* adalah *random noise* dengan *noise power* sebesar 750 dan *sample time noise* sebesar 1, noise ini diletakkan sebelum *integrator* (pada  $\dot{p}$  dan  $\dot{q}$ ), sedangkan noise pada *z* berupa sinyal step dengan *final value* sebesar 1 pada time ke 1. Respon system ditunjukkan pada Gambar 9

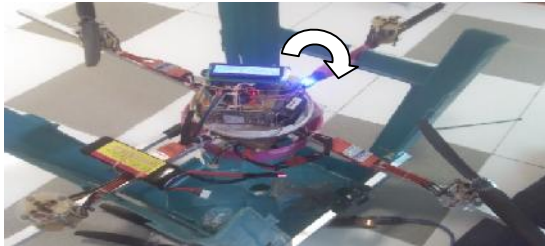


Gambar 9. Respon Sistem Terhadap Noise

Dari Gambar 9 dapat dilihat bahwa dengan menggunakan kontroler *LQR* sistem dapat stabil dan mampu menangani noise, baik yang terjadi pada sudut *roll*, *pitch* maupun ketinggian (*z*).

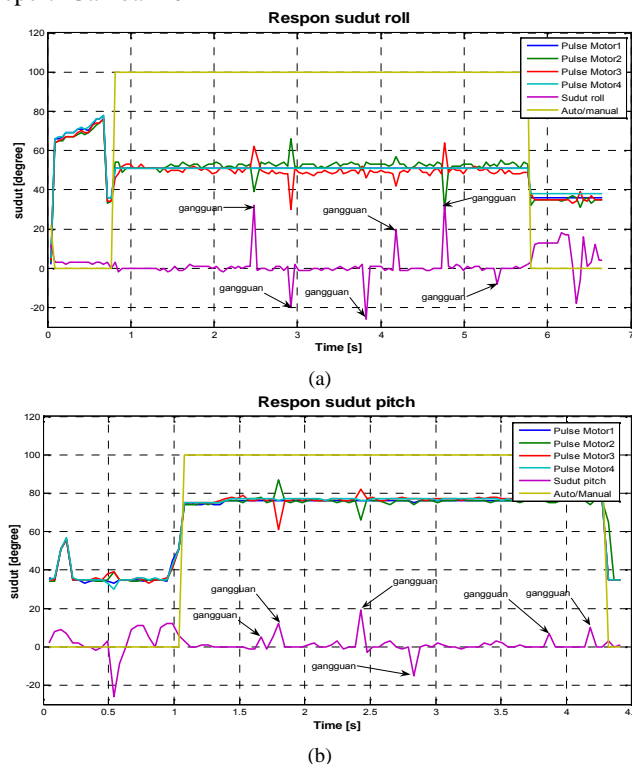
#### D. Implementasi Kontrol *LQR* Pada *Quadcopter*

Kontroler *LQR* yang telah didesain dan diujicoba pada simulasi kemudian diimplementasikan pada plant *quadcopter*. Pengujian pertama dilakukan dengan menguji keseimbangan sudut *roll* dan *pitch* saat *quadcopter* belum diterbangkan, dengan menggunakan alat bantu sederhana, sebagaimana Gambar 10



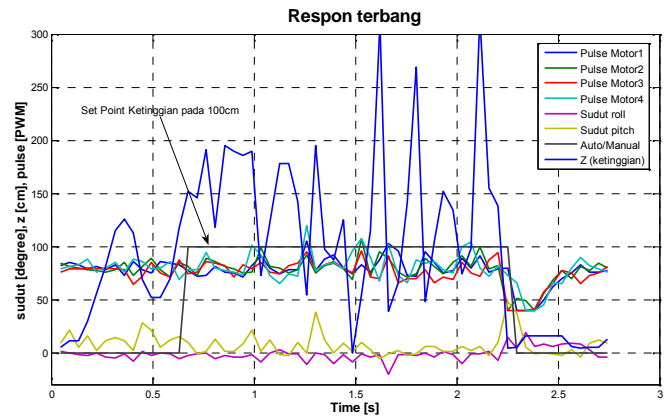
Gambar 10 Uji Kestabilan Sudut Roll dan Sudut Pitch

Gambar 11 menunjukkan respon control *LQR* yang telah ditanamkan pada plant *quadcopter* dan diuji dengan alat bantu seperti Gambar 10



Gambar 10 Respon Sudut Roll (a) dan Sudut Pitch (b) Terhadap Gangguan

Setelah diperoleh respon sudut yang baik terhadap gangguan barulah ditambahkan dengan kontrol ketinggian (*z*), kemudian diujicoba terbang, respon kontrol saat terbang ditunjukkan pada Gambar 11.



Gambar 11 Respon Saat Terbang

Dari respon terbang *quadcopter* tersebut diperoleh  $\tau$  sebesar 0,22 dan *rise time* dengan kriteria 5% - 95% adalah 0,65 detik. Kestabilan sudut pada implementasi ini masih sangat baik, dimana kontroler *LQR* mampu menjaga agar sudut tetap pada 0 derajat. Namun untuk kontrol ketinggiannya dengan *setpoint* pada ketinggian 100cm, masih beresilasi antara 50cm-300cm, hal ini disebabkan oleh pembacaan (*Ping*)) yang membutuhkan waktu yang cukup lama untuk mendapatkan data ketinggian *quadcopter* dengan *range* maksimal 300cm, lama pembacaan (*Ping*)) ini adalah 18,5 milidetik dan pembacaan sensor (*Ping*)) ini juga menyebabkan mikrokontroler bekerja berat. Hal ini mempengaruhi pulsa yang diberikan ke motor sehingga respon yang dihasilkannya menjadi lambat.

#### V. KESIMPULAN/RINGKASAN

Dari hasil penelitian yang telah dilakukan, maka dapat diperoleh beberapa kesimpulan diantaranya:

1. *Complementary filter* yaitu gabungan antara *filter* sensor *accelerometer* dengan *filter* sensor *gyro* memberikan hasil *filter* yang cepat dan akurat.
2. Pada simulasi pengujian untuk Q1 diperoleh  $t_r$  sebesar 2,919s sedangkan dengan memberikan bobot *Q* yang besar pada Q4 diperoleh  $t_r$  hanya 0,1332s, hal ini menunjukkan semakin besar pembobot matrik *Q* semakin cepat pula respon mencapai *steady state*.
3. Pada simulasi kontrol *LQR* mampu mengatasi gangguan yang diberikan baik gangguan pada ketinggian (*z*), sudut *roll* maupun pada sudut *pitch*.
4. Kontrol *LQR* telah berhasil diimplementasikan pada *quadcopter* dalam rangka menjaga kestabilan *hover* dengan hasil yang cukup baik. Namun respon yang dihasilkan tidak sebaik saat simulasi, hal ini dikarenakan kemampuan mikrokontroler dalam mengolah algoritma aritmatika tidak secepat pengolahan pada matlab.

#### DAFTAR PUSTAKA

- [1]. Tommaso Bresciani, "Modelling, Identification and Control of a Quadcopter Helicopter". Department of Automatic Control Lund University, (2008, Oct).

- [2]. Luukkonen, Teppo, “*Modelling and control of quadcopter*”, Aalto University, Espoo, (2011).
- [3]. Jorge Miguel Brito Domingues, “*Quadcopter prototype*”, Grau de Mestre em Engenharia Mecânica, (2009, Oct).
- [4]. Marcelo De Lellis Costa de Oliveira, “*Modeling, Identification and Control of a Quadcopter Aircraft*”, Czech Technical University in Prague, (2011).
- [5]. Dr J. F. Whidborne, “*Modelling And Linear Control Of A Quadcopter*” Cranfield University, (2007).
- [6]. Randal W. Beard , “*Quadcopter Dynamics and Control*”, Brigham Young University, (2008, Oct).
- [7]. Darmawan, Aria “*Perancangan Embedded Kontroler LQR Adaptive Menggunakan Mikrokontroler Untuk Pengaturan Kecepatan Motor DC*”, *Tugas Akhir*, ITS Surabaya, (2011).
- [8]. Gamayanti, Nurlita, “*Karakteristik Sistem Ordo Pertama*” Teknik Sistem Pengaturan, Jurusan Teknik Elektro FTI-ITS, Surabaya, (2010).
- [9]. Naidu, Desineni Subbaram, “*Optimal Control System*”, CRC Press LLC, USA, (2003).
- [10]. Colton, Shane “*The Balance Filter: A Simple Solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing Platform*”, MIT